Support Vector Machines

- Seminar: Regularisierungstechniken und strukturierte Regression -

Giuseppe Casalicchio
Betreuer: Wolfgang Pößnecker

Institut für Statistik, LMU München

15. Januar 2013

Gliederung

Gliederung

Grundidee

Ausgangslage:

N Trainingsdaten $(\mathbf{x}_1^\top, y_1), \ldots, (\mathbf{x}_N^\top, y_N)$ mit

$\mathbf{x}_i^\top \in \mathbb{R}^p$:	Merkmalsvektor mit p Variablen
$y_i \in \{-1, 1\}$:	Klassenzugehörigkeit der i-ten Beobachtung

Ziel:

optimale Zuordnung neuer Daten $\mathbf{x}_{neu}$ in die Klasse $y_{neu} \in \{-1, 1\}$.

Grundidee:

Finde eine Hyperebene, die die Daten möglichst gut in zwei Klassen trennt.

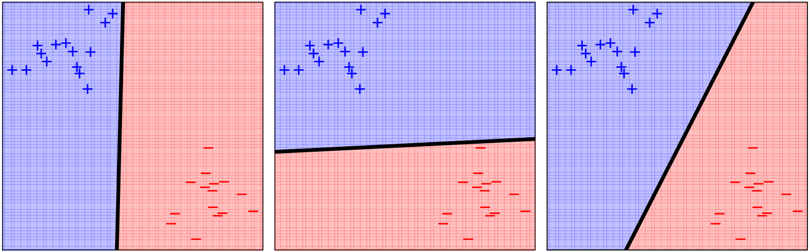
Gliederung

Linear trennbare Daten

Hyperebene

Gesucht: Entscheidungsfunktion $f : \mathbb{R}^p \to \{-1, 1\}$, sodass
$f(\mathbf{x}_i) = y_i \qquad \forall i = 1, \ldots, N$

Frage: Wie wählt man die Hyperebene aus?



Linear trennbare Daten

Hyperebene

Eine Hyperebene trennt einen p-dimensionalen Variablenraum in zwei Unterräume und hat selbst die Dimension $(p-1)$:

$$\{\mathbf{x} \in \mathbb{R}^p \mid \mathbf{w}^\top \mathbf{x} + b = 0\}$$

mit

$\mathbf{w} \in \mathbb{R}^p$:	Vektor orthogonal zur Hyperebene
$b \in \mathbb{R}$:	Verschiebung (vom Ursprung)

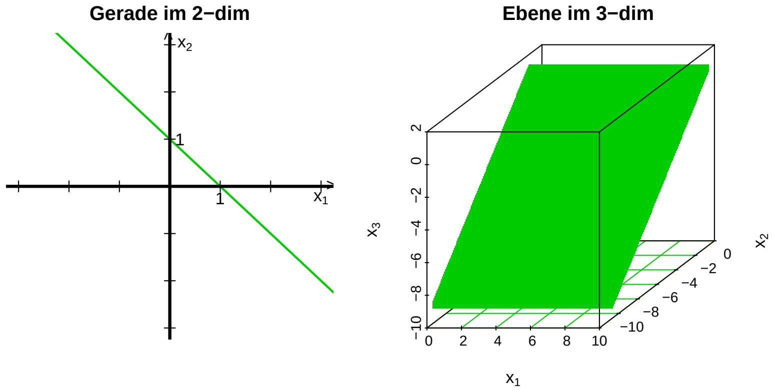
Notation zum Skalarprodukt: $\mathbf{w}^\top \mathbf{x} = \langle \mathbf{w}, \mathbf{x} \rangle = \sum_{i=1}^{p} w_i x_i$

Linear trennbare Daten

Hyperebene

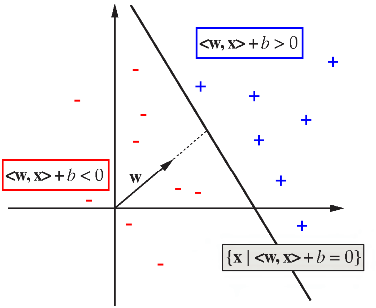
Beispiel: $x_2 = -x_1 + 1 \Leftrightarrow x_1 + x_2 - 1 = 0$
mit $\mathbf{w}^\top = (1, 1)$ und $b = -1$

$\rightarrow$ Gerade im 2-dimensionalen Variablenraum (links).



Linear trennbare Daten

Hyperebene



- Punkte "unterhalb" der Hyperebene $\to -1$
- Punkte "überhalb" der Hyperebene $\to +1$

$$\begin{array}{lll} \text{Klassifiziere in Klasse } -1 & \text{falls} & \mathbf{w}^\top \mathbf{x} + b < 0 \\ \text{Klassifiziere in Klasse } +1 & \text{falls} & \mathbf{w}^\top \mathbf{x} + b > 0 \end{array}$$

$\Rightarrow$ Verwende Entscheidungsfunktion $f(\mathbf{x}) = \mathrm{sgn}(\mathbf{w}^\top \mathbf{x} + b)$

Linear trennbare Daten

Hyperebene

Die Hyperebene, kann gleichermaßen durch alle Paare $\{\lambda \mathbf{w}, \lambda b\},\ \lambda \in \mathbb{R}^+$ dargestellt werden:

$$
\begin{aligned}
&\mathbf{w}^\top \mathbf{x} + b = 0 \\
\Leftrightarrow &\lambda \mathbf{w}^\top \mathbf{x} + \lambda b = 0,\ \text{für } \lambda \in \mathbb{R}^+
\end{aligned}
$$

Problem: Keine eindeutige Beschreibung der Hyperebene

Linear trennbare Daten

Hyperebene

Lösung: Einführung einer kanonischen Hyperebene, für die gilt:

$$\min_{i=1,\ldots,N} |\mathbf{w}^\top \mathbf{x}_i + b| = 1,$$

d.h. nähester Punkt zur Hyperebene hat funktionalen Abstand 1.

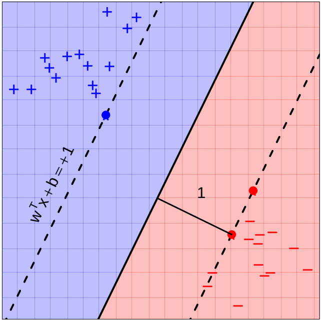
Es gilt:

$$\mathbf{w}^\top \mathbf{x}_i + b \geq +1 \quad \text{für} \quad y_i = +1$$

$$\mathbf{w}^\top \mathbf{x}_i + b \leq -1 \quad \text{für} \quad y_i = -1$$

beziehungsweise

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 \ \forall i = 1, \ldots, N$$



Linear trennbare Daten

Margin

Euklidischer Abstand eines Punktes $\mathbf{x}_i$ zur Hyperebene $(\mathbf{w}, b)$ durch normieren mit der Vektorlänge $||\mathbf{w}||$ bestimmbar:

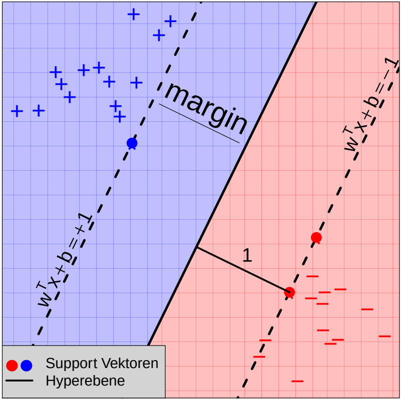
$$d((\mathbf{w}, b), \mathbf{x}_i) = \frac{y_i(\mathbf{w}^\top \mathbf{x}_i + b)}{||\mathbf{w}||} \geq \frac{1}{||\mathbf{w}||}$$

Gesucht: Hyperebene mit größtmöglichen euklidischen Abstand zu den nähesten Punkten.

- → Je kleiner $||\mathbf{w}||$, desto größer der euklidische Abstand.
- → Je größer der euklidische Abstand, desto breiter der Rand (margin).

Linear trennbare Daten

Margin



- Punkte am nähesten zur Hyperebene haben betragsmäßig einen euklidischen Abstand von $\frac{1}{||\mathbf{w}||}$
- Margin (Rand) ist $\frac{2}{||\mathbf{w}||}$ breit
- Alle anderen Punkte liegen jenseits des Randes

Linear trennbare Daten

Primäres Optimierungsproblem

maximiere Rand (margin) $\Leftrightarrow$ minimiere $||\mathbf{w}||$ $\Leftrightarrow$ minimiere $\frac{1}{2}||\mathbf{w}||^2$

$$\min_{\mathbf{w},b} \quad \frac{1}{2}||\mathbf{w}||^2$$

$$\text{NB:} \quad y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 \quad \forall i = 1, \ldots, N$$

Lösung: Lagrange Methode mit Lagrange Multiplikatoren $\alpha_i \geq 0,\ \forall i = 1, \ldots, N$

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}||\mathbf{w}||^2 + \sum_{i=1}^{N} \alpha_i (1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b))$$

$$= \frac{1}{2}||\mathbf{w}||^2 - \sum \alpha_i (y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1)$$

Linear trennbare Daten

$L(\mathbf{w}, b, \boldsymbol{\alpha})$ wird bezüglich

1. b und $\mathbf{w}$ (Primärvariablen) minimiert $\rightarrow \min_{\mathbf{w},b} L(\mathbf{w}, b, \boldsymbol{\alpha})$
2. $\boldsymbol{\alpha}$ (duale Variable) maximiert $\rightarrow \max_{\boldsymbol{\alpha}} (\min_{\mathbf{w},b} L(\mathbf{w}, b, \boldsymbol{\alpha}))$

Zu 1.:

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial b} = 0 \Rightarrow \sum_{i=1}^{N} \alpha_i y_i = 0 \tag{1}$$

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i \tag{2}$$

$\rightarrow$ $\mathbf{w}$ ist Linearkombination der Trainingsdaten $\mathbf{x}_i$.

Linear trennbare Daten

Duales Optimierungsproblem

Zu 2.:
Durch Einsetzen der Lösungen (1) und (2) in $L(\mathbf{w}, b, \boldsymbol{\alpha})$ verschwinden die Primärvariablen $\Rightarrow$ duales Optimierungsproblem:

$$\max_{\boldsymbol{\alpha}} \left(\min_{\mathbf{w},b} L(\mathbf{w}, b, \boldsymbol{\alpha}) \right) = \max_{\boldsymbol{\alpha}} \left(\sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j \right)$$

$$\text{NB:} \quad \alpha_i \geq 0, \qquad \sum_{i=1}^{N} \alpha_i y_i = 0 \qquad \forall i = 1, \ldots, N$$

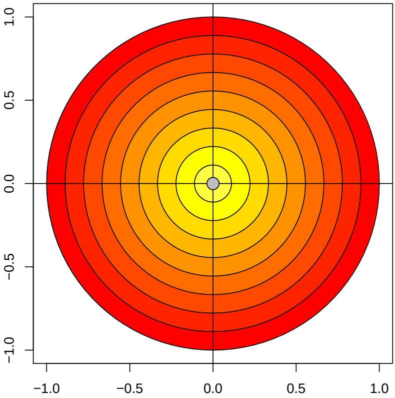
30

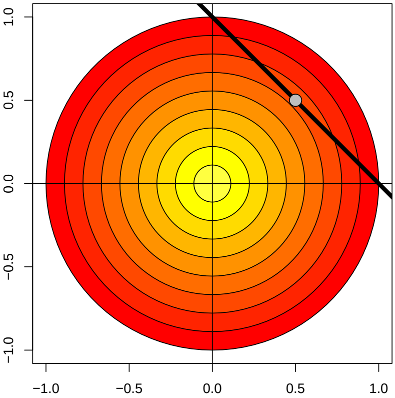
Einschub: Lagrange Methode

$$\min_{x_1,x_2} x_1^2 + x_2^2$$

$$\min_{x_1,x_2} x_1^2 + x_2^2$$

$$\text{NB: } x_1 + x_2 = 1$$

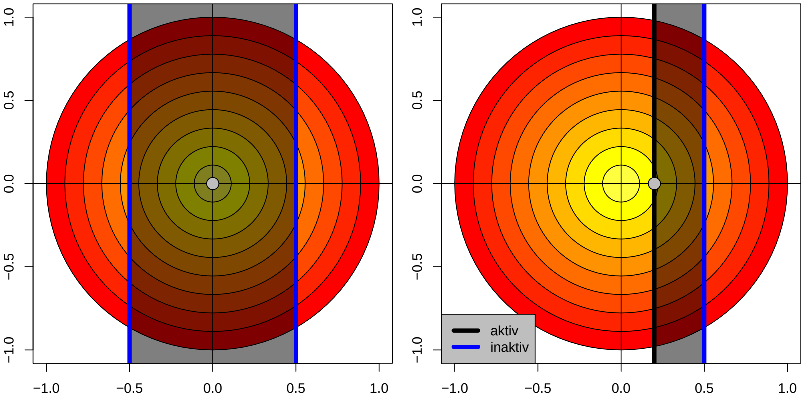




Einschub: Lagrange Methode

$$\min_{x_1, x_2} x_1^2 + x_2^2 \qquad \min_{x_1, x_2} x_1^2 + x_2^2$$

NB: $x_1 \geq -0.5,\ x_1 \leq 0.5$ NB: $x_1 \geq 0.2,\ x_1 \leq 0.5$



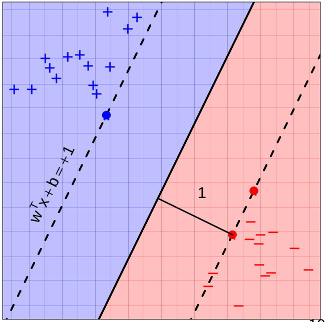
Linear trennbare Daten

Lagrange Methode

Nach Karush-Kuhn-Tucker (KKT) ist eine weitere Bedingung nötig:

$$\alpha_i[y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1] = 0 \quad \forall i = 1, \ldots, N$$

- $\Rightarrow$ $\alpha_i = 0$ für $y_i(\mathbf{w}^\top \mathbf{x}_i + b) > 1$ (funktionale Abstand einer Beobachtung $\mathbf{x}_i$ ist größer 1). $\rightarrow$ inaktive NB
- $\Rightarrow$ Es interessieren nur die support Vektoren $\rightarrow y_i(\mathbf{w}^\top \mathbf{x}_i + b) = 1$



Linear trennbare Daten

Zusammenfassung

1. Lagrangefunktion aufstellen und duale Funktion herleiten
2. Bestimme α_i der support Vektoren durch duales Optimierungsproblem
3. Bestimme $\mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i$ der kanonischen Hyperebene
4. Bestimme Verschiebung $b = -\frac{1}{2}(\mathbf{w}^\top \mathbf{x}^+ + \mathbf{w}^\top \mathbf{x}^-)$ der Hyperebene aus support Vektoren $\mathbf{x}^+$ und $\mathbf{x}^-$:

$$b + \mathbf{w}^\top \mathbf{x}^+ = +1$$
$$b + \mathbf{w}^\top \mathbf{x}^- = -1$$

5. Entscheidungsfunktion
$f(\mathbf{x}) = \operatorname{sgn}\left(\mathbf{w}^\top \mathbf{x} + b\right) = \operatorname{sgn}\left(\sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i^\top \mathbf{x} + b\right)$

Gliederung

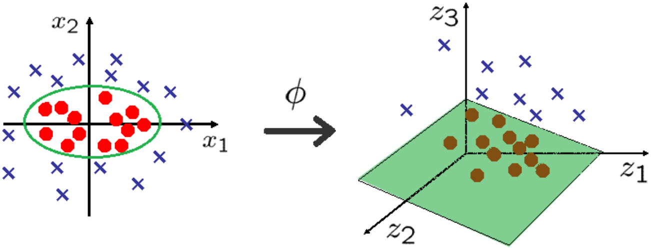
Kern Trick

Idee

Daten in höher dimensionalen Raum überführen, in dem sie linear Trennbar sind.

Beispiel:

$$\begin{array}{rccccl} \Phi: & \mathbb{R}^2 & \rightarrow & \mathbb{R}^3 & & \\ & (x_1, x_2) & \rightarrow & (z_1, z_2, z_3) & = & (x_1^2, \sqrt{2}x_1x_2, x_2^2) \end{array}$$



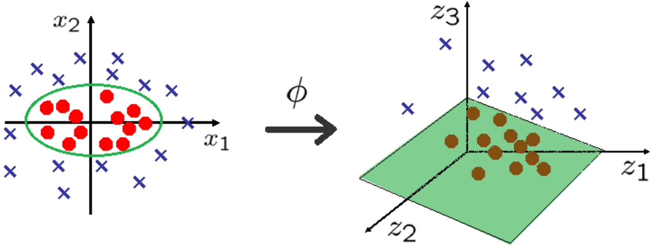
Kern Trick

Beispiel

Die trennende Hyperebene im $\mathbb{R}^3$ hat die Form

$$\begin{aligned} & \mathbf{w}^\top \mathbf{z} + b = 0 \\ \Leftrightarrow\ & w_1 z_1 + w_2 z_2 + w_3 z_3 + b = 0 \\ \Leftrightarrow\ & w_1 x_1^2 + w_2 \sqrt{2} x_1 x_2 + w_3 x_2^2 + b = 0 \end{aligned}$$

$\Rightarrow$ Gleichung einer Ellipse im $\mathbb{R}^2$, da $(z_1, z_2, z_3) = (x_1^2, \sqrt{2} x_1 x_2, x_2^2)$.



Kern Trick

Kernfunktion

$$\text{Bisher:} \qquad f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{N} y_i \alpha_i \langle \mathbf{x}_i, \mathbf{x} \rangle + b\right)$$

$$\text{Jetzt:} \qquad f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{N} y_i \alpha_i \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}) \rangle + b\right)$$

- Φ überführt den Variablenraum in einen höherdimensionalen Variablenraum $\mathcal{M}$
- Trainingsdaten sind in $\mathcal{M}$ linear trennbar
- Φ wird durch eine Kernfunktion $K(\mathbf{x}_i, \mathbf{x})$ festgelegt
- K verhält sich wie ein Skalarprodukt in $\mathcal{M}$:

$$K(\mathbf{x}_i, \mathbf{x}) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}) \rangle$$

Kern Trick

Wichtige Kernfunktionen

- Polynomial: $K(\mathbf{x}_i, \mathbf{x}_j) = (c + \langle \mathbf{x}_i, \mathbf{x}_j \rangle)^d$, für c konstant
- Radial Basis: $K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||}{c}\right)$ für $c > 0$

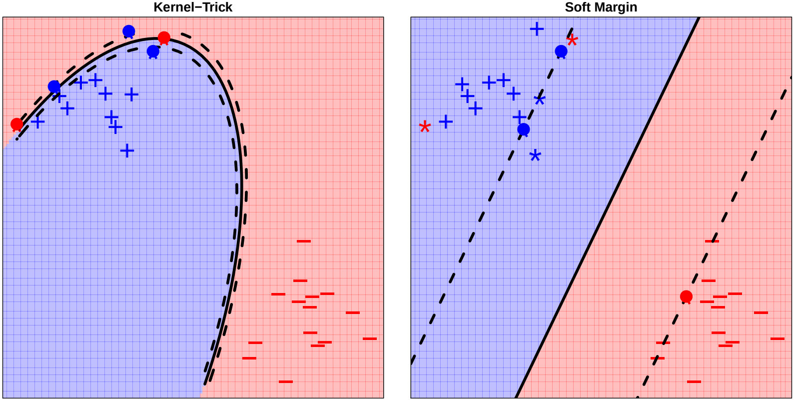
Beispiel: $\mathbf{x}_i = (x_{i_1}, x_{i_2})$, $c = 0$, $d = 2$

$$
\begin{aligned}
K(\mathbf{x}_1, \mathbf{x}_2) &= (\langle \mathbf{x}_1, \mathbf{x}_2 \rangle)^2 = (\langle (x_{1_1}, x_{1_2}), (x_{2_1}, x_{2_2}) \rangle)^2 \\
&= (x_{1_1} x_{2_1} + x_{1_2} x_{2_2})^2 \\
&= (x_{1_1}^2 x_{2_1}^2 + x_{1_2}^2 x_{2_2}^2 + 2 x_{1_1} x_{1_2} x_{2_1} x_{2_2}) \\
&= \langle (x_{1_1}^2, x_{1_2}^2, \sqrt{2} x_{1_1} x_{1_2}), (x_{2_1}^2, x_{2_2}^2, \sqrt{2} x_{2_1} x_{2_2}) \rangle \\
&= \langle \Phi(\mathbf{x}_1), \Phi(\mathbf{x}_2) \rangle
\end{aligned}
$$

Soft Margin

Idee

Bisher: Einzelne Außreiser beeinflussen Hyperebene (Overfitting)
Jetzt: Erlaube Fehlklassifizierung, aber bestrafe diese!



Soft Margin

Erlaube Fehlklassifizierung

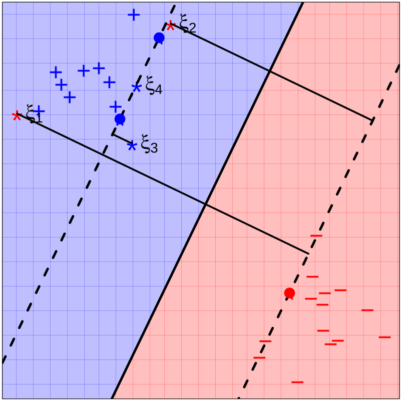
Nebenbedingung durch Schlupfvariablen $\xi_i \geq 0$ lockern, sodass:

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i$$

Die Trainingsdaten sind für:

- $\xi_i = 0$ richtig klassifiziert
- $0 < \xi_i \leq 1$ richtig klassifiziert (innerhalb des Randes)
- $\xi_i > 1$ fehlklassifiziert

$$\xi_i = \max\{0, 1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b)\}$$



Soft Margin

Primäres Optimierungsproblem

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \quad \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{N}\xi_i$$

$$\text{NB:} \quad \begin{aligned} y_i(\mathbf{w}^\top\mathbf{x}_i + b) &\geq 1 - \xi_i \\ \xi_i &\geq 0 \qquad \forall i = 1, \ldots, N \end{aligned}$$

Kompromiss:

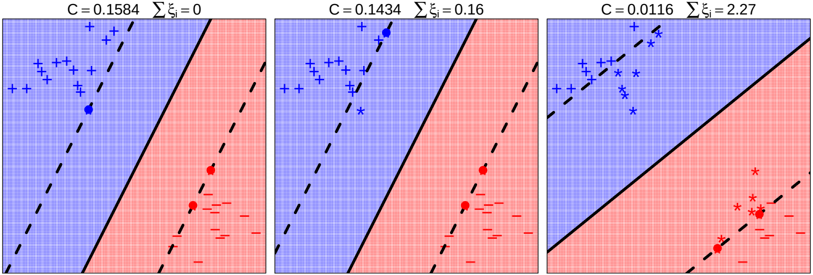
maximiere Rand ($\min \frac{1}{2}||\mathbf{w}||^2$) $\leftrightarrow$ minimiere Trainingsfehler $\sum_{i=1}^{N}\xi_i$

$$L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\mu}) = \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i(y_i(\mathbf{w}^\top\mathbf{x}_i + b) - (1 - \xi_i)) - \sum_{i=1}^{N}\mu_i\xi_i$$

Soft Margin

Parameter C kann durch Kreuzvalidierung bestimmt werden und steuert wie stark Trainingsfehler bestraft werden:

- C groß: korrekte Klassifizierung der Trainingsdaten wichtiger $\rightarrow$ kleiner Rand
- C klein: breiter Rand wichtiger $\rightarrow \sum_{i=1}^{N} \xi_i$ größer



Soft Margin

Duales Optimierungsproblem

Minimierung der Lagrangefunktion bezüglich $\mathbf{w}$, b und $\boldsymbol{\xi}$ und Einsetzen der Lösungen in das primäre Optimierungsproblem führt zum dualen Optimierungsproblem (vgl. Folie 16):

$$\max_{\boldsymbol{\alpha}} \left(\min_{\mathbf{w},b,\boldsymbol{\xi}} L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\mu}) \right) = \max_{\boldsymbol{\alpha}} \left(\sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j \right)$$

$$\text{NB:} \qquad 0 \leq \alpha_i \leq C, \qquad \sum_{i=1}^{N} \alpha_i y_i = 0 \qquad \forall i = 1, \ldots, N$$

Weiteres Vorgehen analog zum linear trennbaren Fall.

Gliederung

Zusammenfassung und Ausblick

Zusammenfassung

Linear trennbare Daten:

- Aufstellen der Hyperebenengleichung
- Bestimmung der Hyperebenenparameter durch maximierung des Randes (→ Lagrange-Methode)

Nicht linear trennbare Daten:

1. **Kern Trick:**
   - In höherdimensionalen Variablenraum überführen, in dem Trainingsdaten linear trennbar sind.
   - Bestimmung der Hyperebenenparameter analog.
2. **Soft Margin:**
   - Erlaube Fehlklassifikation, aber bestrafe diese.
   - Gleiches duales Optimierungsproblem mit zusätzlicher NB.
     → Vorgehensweise wie bei linear trennbaren Daten.
3. **Kern Trick und Soft Margin**

Zusammenfassung und Ausblick

Ausblick

- Erweiterung für Regressionsprobleme
- Erweiterung für mehrkategorialen Response, z.B. durch **Paarweise Klassifikation:**
  - Bilde Klassifikatoren für jedes mögliche Paar der K Klassen
  - Zuordung neuer Beobachtungen durch Mehrheitsentscheid in die Klasse $k \in \{1, \ldots, K\}$

Beispiel:

Für $K = 3$ gibt es $\frac{K(K-1)}{2} = 3$ mögliche Paare / Klassifikatoren:

	Beob. 1	Beob. 2
$k \in \{1, 2\}$	1	1
$k \in \{1, 3\}$	1	3
$k \in \{2, 3\}$	2	3
Mehrheitsentscheid	$\rightarrow 1$	$\rightarrow 3$

Gliederung

References I

B. Schölkopf and A. J. Smola
Learning with Kernels: Support vector machines, regularization, optimization, and beyond
Massachussetts Institute of Technology, 2002

J. Friedman, T. Hastie and R. Tibshirani
The elements of statistical learning
Springer Series in Statistics, 2011

S.R. Gunn and others
Support vector machines for classification and regression
ISIS technical report vol. 14, 1998

Rpackage: `e1071` Funktion `svm`

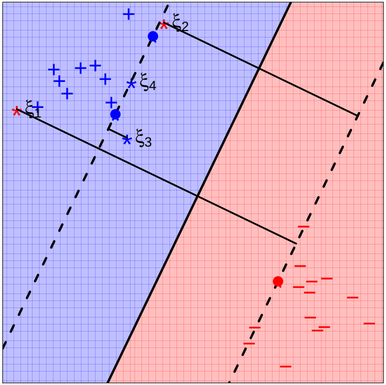
Vielen Dank für die Aufmerksamkeit!

Anhang

Erweiterung auf Regressionsprobleme

$$\min \; \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{N} \xi_i$$

$$\xi_i = \max\{0, 1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b)\}$$
$$= [1 - y_i \underbrace{(\mathbf{w}^\top \mathbf{x}_i + b)}_{f(\mathbf{x}_i)}]_+$$



Alternative **Loss + Penalty** Form:

$$\min \; \sum_{i=1}^{N} [1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b)]_+ + \frac{\lambda}{2}||\mathbf{w}||^2 \text{ mit } \lambda = 1/C$$

Verlustfunktion: $L(y_i, f(\mathbf{x}_i)) = [1 - y_i f(\mathbf{x}_i)]_+$ (hinge loss)
$\Rightarrow$ Verwende andere Verlustfunktion für Regressionsprobleme

Anhang

Vor- und Nachteile

Vorteile:

- Nicht nur für Klassifikation geeignet → viele Erweiterungen
- flexiblere Klassifizierung durch Arbeiten in höheren Dimensionen
- Parameterschätzungen basieren auf Teilmenge der Trainingsdaten (support vectors) → schnelle Klassifizierung

Nachteile:

- Geeignete Kernfunktion muss empirisch gesucht werden
- Geeignete wahl für C → muss empirisch gesucht werden
- Erweiterungen teilweise aufwendig oder ineffizient (z.B. Paarweise Klassifikation)

Anhang

Herleitung duale Funktion bei linear trennbare Daten

Mit (1) und (2) von Folie 15 lässt sich zeigen:

$$
\begin{aligned}
L(\mathbf{w}, b, \boldsymbol{\alpha}) &= \frac{1}{2}\mathbf{w}^\top \mathbf{w} - \sum_{i=1}^{N} \alpha_i (y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1) \\
&\overset{(2)}{=} \frac{1}{2} \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i^\top \sum_{j=1}^{N} \alpha_j y_j \mathbf{x}_j - \sum_{i=1}^{N} \alpha_i y_i \left(\sum_{j=1}^{N} \alpha_j y_j \mathbf{x}_j^\top \right) \mathbf{x}_i - \sum_{i=1}^{N} b \alpha_i y_i + \sum_{i=1}^{N} \alpha_i \\
&\overset{(1)}{=} \frac{1}{2} \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i^\top \sum_{j=1}^{N} \alpha_j y_j \mathbf{x}_j - \sum_{i=1}^{N} \alpha_i y_i \left(\sum_{j=1}^{N} \alpha_j y_j \mathbf{x}_j^\top \right) \mathbf{x}_i + \sum_{i=1}^{N} \alpha_i \\
&= -\frac{1}{2} \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i^\top \sum_{j=1}^{N} \alpha_j y_j \mathbf{x}_j + \sum_{i=1}^{N} \alpha_i
\end{aligned}
$$

Anhang

Herleitung duale Funktion bei nicht linear trennbare Daten

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\mu})}{\partial \boldsymbol{\xi}} = 0 \Rightarrow C = \alpha_i + \mu_i \qquad \forall i = 1, \ldots, N \tag{3}$$

$$L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\mu}) = \frac{1}{2}\mathbf{w}^\top \mathbf{w} + C \sum_{i=1}^{N} \xi_i$$

$$- \sum_{i=1}^{N} \alpha_i (y_i(\mathbf{w}^\top \mathbf{x}_i + b) - (1 - \xi_i)) - \sum_{i=1}^{N} \mu_i \xi_i$$

$$\overset{(2)}{=} \frac{1}{2} \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i^\top \sum_{j=1}^{N} \alpha_j y_j \mathbf{x}_j + \cancel{\sum_{i=1}^{N} \overbrace{(\alpha_i + \mu_i)}^{(3) \Rightarrow C}} \xi_i$$

$$- \sum^{N} \alpha_i y_i \left(\sum^{N} \alpha_j y_j \mathbf{x}_j^\top \right) \mathbf{x}_i - \cancel{\sum^{N} b}\, \alpha_i y_i + \sum^{N} \alpha_i \cancel{- \sum^{N} \alpha_i \xi_i - \sum^{N} \mu_i \xi_i}$$